

Introduction

0.1 Brief introduction on the STEP standard

STEP is the casual name for the ISO 10303 standard and stands for Standard for the Exchange of Product model data. Work on the standard started in 1984 and is still in progress with participants from all over the world. This introduction is compiled using elements from [STEP-1], [Kuiper-94] and [Fowler-95].

0.1.1 The objectives of STEP

STEP part 1 says: "ISO 10303 is a series of International Standards for the computer-interpretable representation and exchange of product data. The objective is to provide a neutral mechanism capable of describing product data throughout the life cycle of a product, independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also a basis for implementing and sharing product databases and archiving." [STEP-1].

In other words the STEP methodology has been developed to meet industry requirements for standard data specifications that support:

- long term storage and retention of product information;
- reduction and elimination of "islands of automation";
- independence of data from the software tools which create or consume information;
- communication of product information between departments, disciplines, and enterprises.

In addition, the fact that STEP is a standard introduces additional requirements, in that the specification developed to fulfil these requirements should be stable, extensible, and publicly available.

0.1.2 Principles

The STEP methodology is based on a small number of fundamental principles:

- STEP defines an architecture for product data, providing stability and extensibility;
- STEP supports and requires traceability of data to industry needs;
- The role of STEP is the standardisation of industry application semantics;
- STEP defines the requirements for implementation of product data exchange, based on a separation of data specifications (the logical definition) from implementation forms (the physical realisation);
- STEP defines the requirements for the assessment of conformance of implementations.

0.1.3 The STEP architecture

The architecture of STEP results from the principles stated above. The complete architecture of STEP is described in the *ISO 10303 Architecture and Reference Manual*, which is currently being prepared for publication. Below the main elements of the architecture will be highlighted.

Layered approach with functional series of standard documents

The STEP standard takes a layered approach to product data modelling. The different layers are called series and these contain parts which are related by the function they perform. The series are given in Table 1.

Each series adds specific aspects to the product data model description. The higher numbered layers build further on concepts defined in the lower numbered layers. The EXPRESS language is used to formally define the data models (*schemas* in STEP parlance) of the parts in layers 4, 5, 6 and 7. Application protocols are directed for use by a specific user community, engineering discipline or branch of industry. The standard also includes a formal methodology for conformance testing of STEP-based products that claim to implement one of the application protocols.

Table 1 - The series of STEP standard documents

<i>part numbers</i>	<i>series title</i>	<i>example documents</i>
<i>10 series</i>	Description Methods	ISO 10303 Part 11 The EXPRESS Language Reference Manual
<i>20 series</i>	Implementation Methods	ISO 10303 Part 21 Clear Text Encoding of the Exchange Structure ISO 10303 Part 22 Standard Data Access Interface
<i>30 series</i>	Conformance Testing Methodology and Framework	ISO 10303 Part 31 General Concepts
<i>40 series</i>	Integrated Generic Resources	ISO 10303 Part 41 Fundamentals of Product Description and Support ISO 10303 Part 42 Geometric and Topological Representation
<i>100 series</i>	Integrated Application Resources	ISO 10303 Part 105 Kinematics
<i>500 series</i>	Application Interpreted Constructs	ISO 10303 Part 509 Manifold Surface Shape Representation
<i>200 series</i>	Application Protocols	ISO 10303 AP 203 Configuration Controlled Design
<i>300 series</i>	Abstract Test Suites	ISO 10303 Part 303 Abstract Test Suite – Configuration Controlled Design (for AP 203)

Key elements of the STEP architecture

Figure 0-1 provides a high level summary of the key elements of the STEP architecture. The arrows in Figure 0-1 specify the *existence dependence* between the elements, i.e. an element at an arrow's tail is dependent on an element at its head.

After the figure a short description of each of the key elements is given.

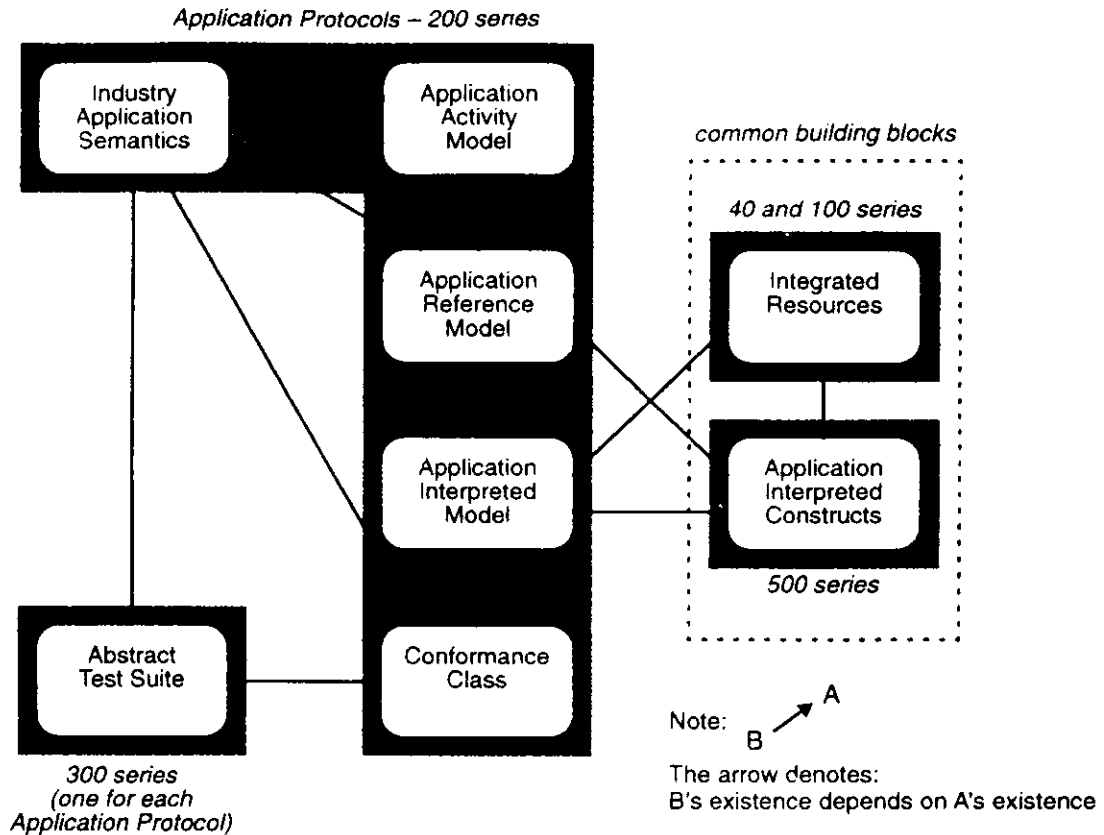


Figure 0-1 - Key elements of the STEP architecture – the grey areas denote the relation between these elements and the STEP documentation structure

Industry Application Semantics

Figure 0-1 shows that the key element upon which all other elements of the STEP architecture depend is industry application semantics. It identifies the processes and data that are essential to an industrial application domain – in other words the industrial needs and semantics that are particular to data exchange in some industry application domain. It must be possible to trace back the concepts and definitions used in the other architectural elements to the needs and semantics in the Industry Application Semantics.

Application Activity Model

Industry application semantics are defined more formally by reference to an Application Activity Model (AAM). This model is created with the IDEF0 activity modelling technique and supports the analysis of the activities and information flows within the industry application. The AAM provides a context model for the application domain and the activities and information flows that are “in scope” or “out of scope” are explicitly declared. The “in scope” information flows can then be elaborated in further detailed STEP analysis and specification models – in particular the Application Reference Model (ARM). It should be noted that the role of the AAM is to capture the activities within an industry application (“what is done”), not the detailed processes (“how it is done”). The latter are likely to vary between organisations, or with time as the result of continuous improvement or business process re-engineering.

Application Reference Model

The second element of the STEP architecture that results from detailed analysis of the product data exchange requirements in the industrial application domain, is the Application Reference Model (ARM). This is a detailed specification of the data objects (entities and attributes) and their relationships, that completely define the “in scope” information flows, which were identified in the AAM. This specification is prepared through analysis of requirements identified by experts in the

industry application – sometimes referred to as “domain experts”. These requirements are therefore described using the terminology of the application domain. They form the basis for further development as well as for review and validation, e.g. by peer experts.

Application Interpreted Model

The data exchange requirements laid down in the ARM are translated – by a “STEP expert” – into a normative STEP data specification in the Application Interpreted Model, as much as possible through selection and constraint of the common building blocks: the Integrated Resources and the Application Interpreted Constructs. This re-use of standard data constructs across a wide range of industry requirements results in a high degree of consistency and integration across models, and enables potential re-use of the software code used in interfaces and the potential sharing of common data across application domains. The AIM is defined using the EXPRESS language, which is computer interpretable and therefore enables file-based exchange according to the STEP Part 21 physical file format and/or data access using the STEP Part 22 SDAI.

Integrated Resources

The Integrated Resources are specified in context independent models and form standard data constructs that are used in the creation of an AIM. They are data models that reflect and support the common requirements of many different product data application areas. Together the Integrated Resources constitute a single, logical, conceptual product data model. They are, however, not themselves intended for direct implementation: they define re-usable building blocks that are intended to be combined and refined to meet specific requirements stated in an ARM. Integrated Resources are specified in EXPRESS.

Application Interpreted Constructs

The Application Interpreted Constructs are a relatively late addition to the STEP architecture. They capture the fulfilment of a collection of requirements that are common to two or more ARMs, i.e. shared by two or more industry application domains. Application Interpreted Constructs are also specified in EXPRESS and explicitly identify the potential for shared data or so-called *interoperability* between two or more Application Protocols / industry application domains.

The difference between Integrated Resources and Application Interpreted Constructs can be explained as follows:

- Integrated Resources were developed – and are still being developed and refined – on the basis of an a priori conception by product data modelling experts of what common product data modelling resources would be needed across a wide range of application domains. As stated before Integrated Resources are not intended for direct implementation in an AIM, but often need to be combined and refined first.
- Application Interpreted Constructs are the common collections of requirements in the ARM and their corresponding implementation in the AIM that are “discovered” during the development of various Application Protocols. Application Interpreted Constructs are a ready-for-implementation kind of “plug-in” modules that can be included in an AIM.

Conformance Classes

An Application Interpreted Model gives the normative specification for data to be exchanged between computer applications. This provides the scope and boundaries for implementations of product data exchange that conform to STEP, and also the scope and boundaries for testing implementations. In order to meet the needs of differing computer systems used within a given industrial application, whilst maintaining consistency of implementation and testing, two or more Conformance Classes may be specified for an AIM. A Conformance Class defines a subset of the AIM that may be used as the basis for implementation and testing. These subsets define the minimum conforming implementation based on the AIM; implementations based on any other subsets are not considered to be conforming.

Abstract Test Suites

The importance of testing and testability within STEP is reflected by a standardised framework and methodology for conformance testing. The Abstract Test Suite is the manifestation of the needs of testing within the STEP architecture. An Abstract Test Suite specifies, in non-specific or parameterised form, the test cases that will be used in assessing the conformance of an implementation to the data specification contained in an AIM and the other elements of the STEP architecture upon which an

AIM depends. Experience in other domains, such as the OSI standard for Open Systems, has shown that standardisation of Abstract Test Suites is an essential prerequisite to repeatability and consistency of testing, and therefore of mutual recognition of test results across regional or national bodies.

0.1.4 The basis of the STEP product data model

The consistency of data specifications within STEP for a wide range of industry applications – Application Protocols – is ensured by the reuse of common Integrated Resources. The Integrated Resources themselves are based in a formalised framework for product data, sometimes referred to as the Generic Product Data Model. The framework defines the basis of all data specification that are standardised within STEP.

Elements of data specifications (or “constructs”) are taken to be the representation of facts about objects in the real world. The Generic Product Data Model framework is based on a classification of the types of data that describe products. The classification identifies five major types of data:

- *The application context* – data that defines the purpose for which product information is created, and the types of product, disciplines, and life-cycle stages for which such information is valid. The use of an application context allows e.g. data that represents an “as designed” product to be distinguished from that for an “as built” configuration, etc.
- *The definition of a product* – data that identifies products, including variants, categories and/or life-cycle “views” of products. Product definition data also includes that which relates to the structure of products, in terms of assembly structures, configurations, effectivities, bill of material, etc.
- *The definition of product properties* – data that characterises products by their properties, independent of the representation of such properties. For example, it is possible to identify the shape of an object, or aspects of the shape, as a property of the object, without providing a detailed description of the shape using a CAD model or an engineering drawing. Examples of product properties are: shape, surface finish, constituting material.
- *The representation of product properties* – data that represents the properties of a product, including multiple representations of the same property. For example, the shape of an object may be identified, and then represented in many different ways: a 3D CAD CSG model, a physical mock-up, an engineering drawing, a technical illustration and a mathematical model are different representations of the same shape.
- *The presentation of the representation of product properties* – data that defines the presentation of product information to support human communication. For example, the shape of an object (which is a product property) is represented by coordinate values, curves, surfaces, etc.; then this representation is presented by assigning colours, line fonts and a viewpoint, and displaying the resulting picture on a workstation.

This classification of product data is the basis for all STEP data specifications. It is the framework upon which all the Integrated Resources are built, and is reflected in the Application Interpreted Models (AIMs) of all Application Protocols. The models that capture this framework embody the principle of existence dependence, which ensures that all product information is related to an identified product and ultimately to an application context. This classification structure is summarised graphically in Figure 0-2.

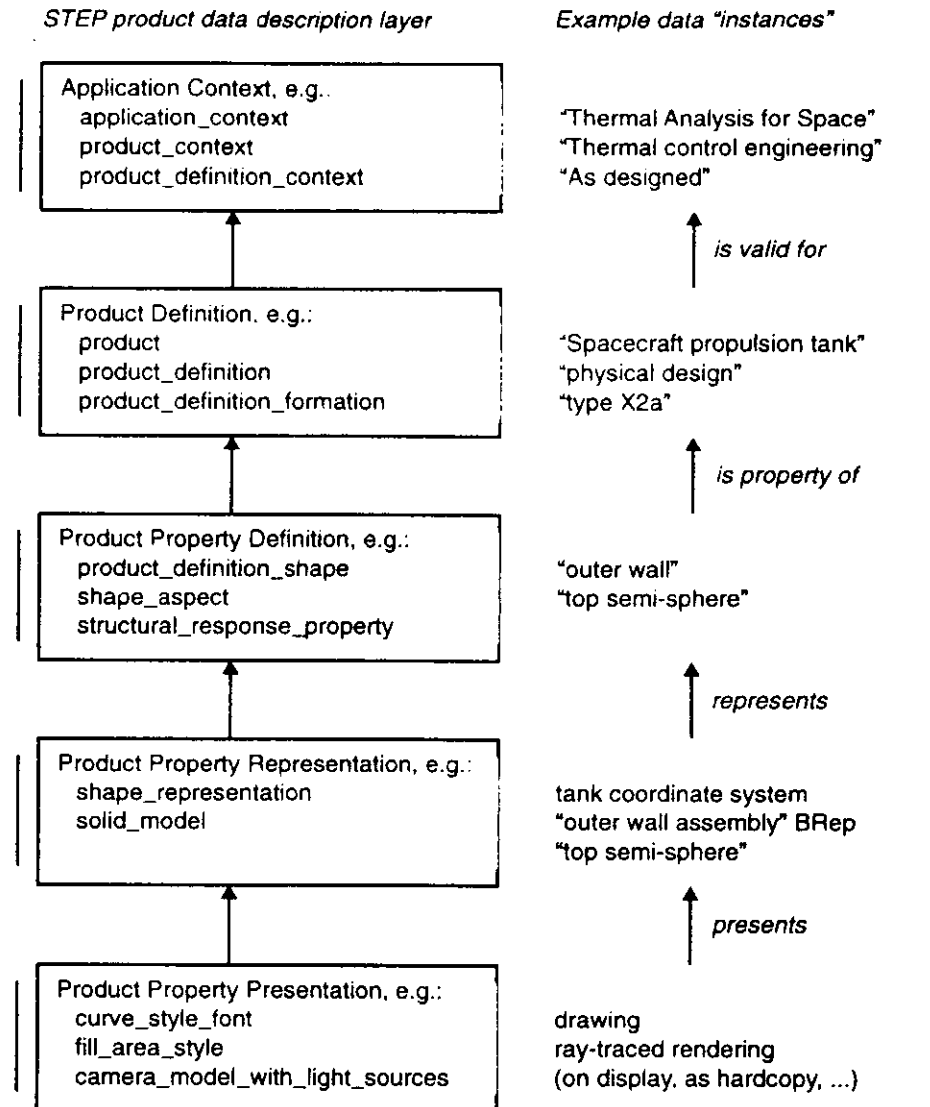


Figure 0-2 - Product data type classification and existence dependence in STEP data models

This principle can lead to models that are at first sight counter-intuitive: rather than stating that a product has a shape, a shape is "of" a product. However, simple analysis of this example shows that the existence dependent form of the model requires that a shape is always the shape of a product. Similarly, at a lower level in the structure, STEP does not allow the existence of "geometry" data just as collections of points, lines, curves, etc. Through the existence dependent structures in the STEP models, such a collection of geometry data must be the representation of some property, that is related to the definition of a product, that has validity in some application context. Thus the basic structure of the STEP models satisfies and enforces the principle identified above: that all product data should be traceable to an industrial need.

0.1.5 Realisation of a STEP solution for product data exchange

One of the main features of the STEP standard is the division between *logical* and *physical* representation of model data. This helps in achieving a clean development process which is best illustrated by the following diagram.

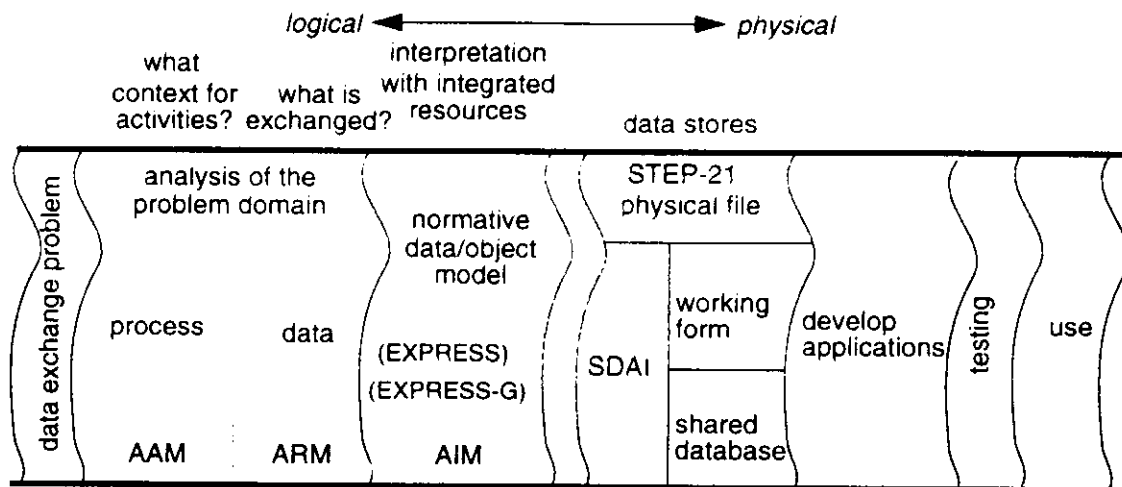


Figure 0-3 - Illustration of the development steps needed to reach a working STEP product data exchange solution and different elements of the STEP standardisation methodology
(adapted from a diagram by P. Kuiper – TNO)

Actual data exchange takes place through the physical implementation in either a [STEP-21] physical file, a so-called working form or shared database. The physical implementation is derived in a standardised way from the logical definition in an Application Interpreted Model (AIM) of a certain Application Protocol and can possibly be accessed through an SDAI [STEP-22] compliant programming interface, for which a significant part could have been computer-generated from the AIM EXPRESS schema by a suitable STEP toolkit.

0.1.6 Status of the STEP standard and its implementations

The STEP standard is becoming mature and the First Release (Parts 1, 11, 21, 31, 41, 42, 43, 44, 46, 101, 201, 203) has been issued per September 1994 as International Standard (IS). Commercial development tools are available on the market and most major CAx software vendors are now offering STEP-based import/export facilities or have announced to support the STEP standard.

0.2 The Industrial Need for STEP-based Data Exchange for Thermal Analysis for Space

The thermal control engineering of spacecraft and other space-related equipment requires extensive computer-supported analyses and tests, at different levels of detail. In typical space projects several companies and/or institutions work together. They are often located at different sites, which may be geographically far apart and they often use different computer platforms. Efficient and reliable electronic exchange of model and results data is therefore highly needed.

Currently, the exchange of such data is done via a great number of dedicated – mostly tool-specific – data formats and the use of many point-to-point converter tools. These tools convert data in one format as well as they can into data in another format. The proliferation of data formats and corresponding converter tools place a great burden on software development and maintenance resources. The converter tools need to be regularly updated as data formats change or get extended at either side of an interface. There is often a loss of information in the conversion process, because finding and implementing the best mapping of concepts between two tool-specific data formats is a non-trivial task and normally a converter developer controls only one of the two data format interfaces.

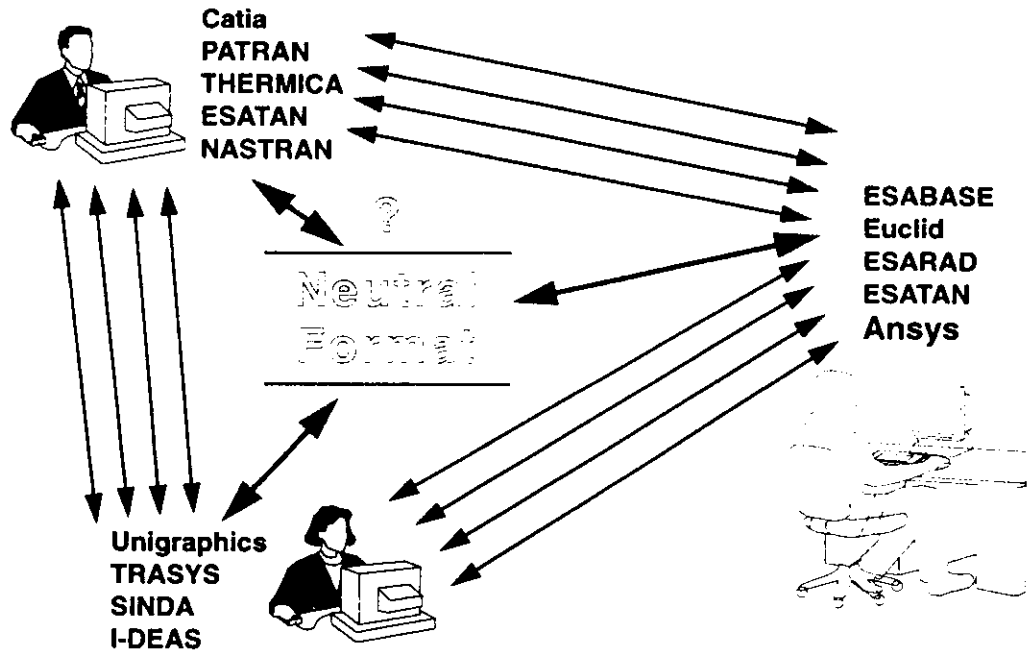


Figure 0-4 - Illustration of the problem of model and data exchange in thermal engineering for space (with an arbitrary selection of tools)

Apart from direct data exchange between e.g. a main contractor and a client or a sub-contractor and its main contractor, there is also a need for a stable neutral format for long term electronic archiving and a growing need to support concurrent engineering: i.e. exchanging models and results with other engineering disciplines during the design and development process, e.g. for thermo-elastic analysis. In the long term, ideally concurrent engineering would be supported by having shared databases which are simultaneously accessible to all engineers of different disciplines in a project team.

The current STEP-TAS Application Protocol attempts to lay the foundation to address these needs.

An important additional possibility of this STEP based protocol is that it is defined in such a generic way that it could be extended in the future to cover not only space thermal analysis, but also the related field of space environmental analysis disciplines.